

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-300522

(43)Date of publication of application : 11.10.2002

(51)Int.Cl.

H04N 5/91

H04N 5/78

H04N 5/92

(21)Application number : 2001-100643

(71)Applicant : SONY CORP

(22)Date of filing : 30.03.2001

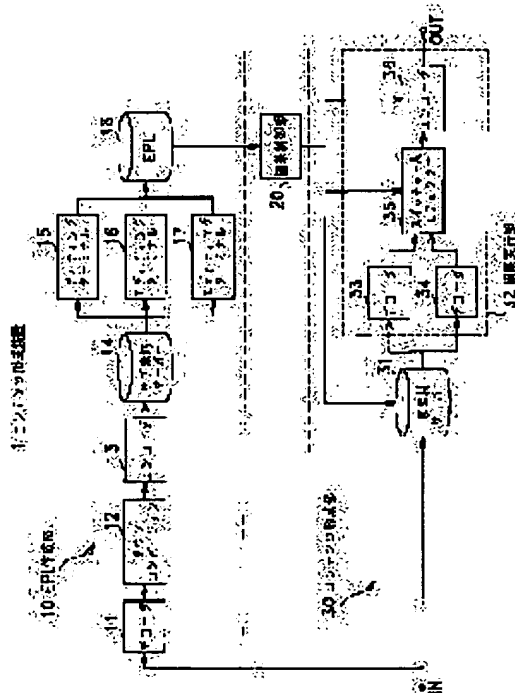
(72)Inventor : MURAKAMI YOSHIHIRO  
KAWAHARA MINORU  
MAGAI MITSUTOSHI  
MUKAI MASATAKA

## (54) EDIT LIST GENERATING METHOD

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a play list generating method which corresponds to the data of a multiformat, while the deterioration of data quality is prevented, can exchange the processing without once accumulating it and using a material and enables formation of contents.

**SOLUTION:** Edit terminals 15, 16 and 17 form the main part of an EPL generation part 10, and read and decode a high compressed sub-material and generate EPL, while a video based on obtained material data is displayed on a monitor. EPL has identification information for specifying the material used for editing and a format declaration statement, defining the format of at least a part of the material. EPL generated by the edit terminals 15, 16 and 17 is stored in an EPL storage part 18. EPL stored in the EPL storage part 18 is read into an edit control part 20. The edit control part 20 generates an edit control signal, based on EPL and supplies the edit control signal to a contents forming part 30.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

This Page Blank (uspto)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-300522

(P2002-300522A)

(43) 公開日 平成14年10月11日 (2002. 10. 11)

(51) Int.Cl.<sup>7</sup>

識別記号

F I

テーマト\* (参考)

H 0 4 N 5/91  
5/78  
5/92H 0 4 N 5/78  
5/91  
5/92B 5 C 0 5 3  
N  
H

審査請求 未請求 請求項の数16 O L (全 22 頁)

(21) 出願番号 特願2001-100643(P2001-100643)

(22) 出願日 平成13年3月30日 (2001. 3. 30)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 村上 芳弘

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72) 発明者 河原 実

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(74) 代理人 100067736

弁理士 小池 晃 (外2名)

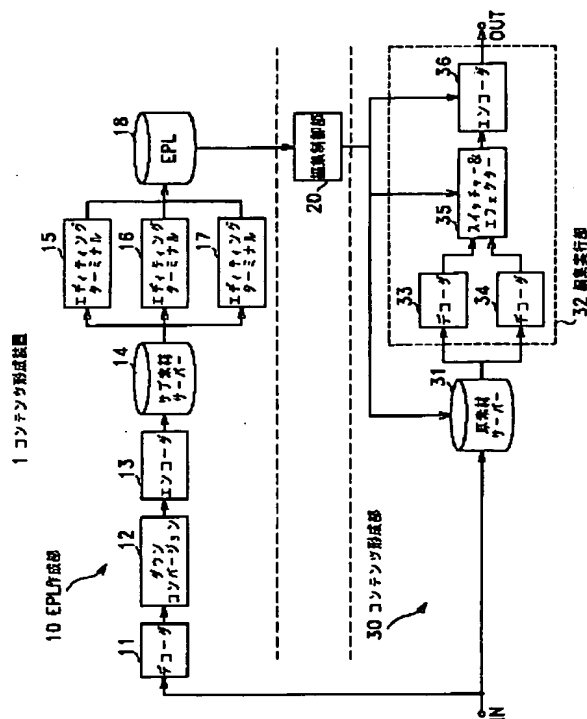
最終頁に続く

(54) 【発明の名称】 編集リスト作成方法

(57) 【要約】

【課題】 データの品質劣化を防ぎながら、またマルチフォーマットのデータに対応し、かつ素材を手許に一旦溜めて使うことなく、さらに処理の互換を可能として、コンテンツを形成することができる編集リストを作成するための編集リスト作成方法を提供する。

【解決手段】 エディティングターミナル15、16、17は、EPL作成部10の主要部をなすものであり、高圧縮サブ素材を読み出して復号し、得られた素材データに基づく映像をモニタに表示しながら、EPLを作成する。EPLは、編集に使用される素材を特定するための識別情報と、少なくとも一部の素材のフォーマットを定義するフォーマット宣言文とを有する。エディティングターミナル15、16、17で作成されたEPLは、EPL記憶部18に格納される。このEPL記憶部18に格納されたEPLは、編集制御部20に読み出される。編集制御部20は、EPLに基づいて編集制御信号を生成し、編集制御信号をコンテンツ形成部30に供給する。



## 【特許請求の範囲】

【請求項1】 複数の映像及び／又は音声の素材を用いてコンテンツを形成するための編集リストを作成する編集リスト作成方法において、編集による出力を識別する識別情報とフォーマットを特定するフォーマット宣言文とを少なくとも用いて編集内容を定義する編集定義工程と、入力素材を識別する識別情報と入力素材のフォーマットを特定するフォーマット宣言文とを少なくとも用いて入力素材を定義しながらその入力素材に施す処理を定義していく処理定義工程とを備えることを特徴とする編集リスト作成方法。

【請求項2】 上記出力の識別情報及び入力素材の識別情報は、グローバルユニークな同一体系の識別コードであることを特徴とする請求項1記載の編集リスト作成方法。

【請求項3】 上記処理定義工程は、入力素材に施す処理方法をグローバルユニークな同一体系の識別コードで記述することを特徴とする請求項1記載の編集リスト作成方法。

【請求項4】 上記処理定義工程は、上記識別情報とフォーマット宣言文とを少なくとも用いて複数の入力素材を定義しながら、各入力素材に施す複数の処理を定義していくことを特徴とする請求項1記載の編集リスト作成方法。

【請求項5】 上記処理定義工程は、各入力素材に施す各処理方法をグローバルユニークな同一体系の識別コードで記述することを特徴とする請求項4記載の編集リスト作成方法。

【請求項6】 上記編集内容を特定するフォーマット宣言文は編集出力映像の画像フォーマットを含み、上記入力素材を特定するフォーマット宣言文は入力映像素材の画像フォーマットを含むことを特徴とする請求項1記載の編集リスト作成方法。

【請求項7】 上記画像フォーマットは、撮影時の1コマの時間、画像の縦横の有効画素数、各画素の縦横比の項目を少なくとも含むスキャンフォーマットであることを特徴とする請求項6記載の編集リスト作成方法。

【請求項8】 上記処理定義工程が定義した入力素材を高圧縮したサブ素材を用いて編集リストを作成することを特徴とする請求項1記載の編集リスト作成方法。

【請求項9】 上記編集定義工程と上記処理定義工程を用いて作成した少なくとも二つの編集リストを統合・最適化して編集リストを作成する統合・最適化工程を備えることを特徴とする請求項1記載の編集リスト作成方法。

【請求項10】 既に作成された他の編集リストに該編集リスト自体を識別するためのグローバルユニークな識別コードを付して、編集リスト内に記述することを許す階層構造の統合編集リストを上記統合・最適化工程を用

いて作成することを特徴とする請求項9記載の編集リスト作成方法。

【請求項11】 上記統合・最適化工程は、上記少なくとも二つの編集リストを用いて階層構造を有する統合最適化編集リストを作成することを特徴とする請求項9記載の編集リスト作成方法。

【請求項12】 上記統合・最適化工程は、上記少なくとも二つの編集リストをネットワークを介して受け取って階層構造を有する統合最適化編集リストを作成することを特徴とする請求項11記載の編集リスト作成方法。

【請求項13】 上記少なくとも二つの編集リストには識別情報を付すことを特徴とする請求項9記載の編集リスト作成方法。

【請求項14】 上記少なくとも二つの編集リストには、上記統合・最適化工程にて上位の編集リストへ受け渡しされるときの位置及び／又は下位の編集リストの受け取り位置が記述されていることを特徴とする請求項9記載の編集リスト作成方法。

【請求項15】 上記少なくとも二つの編集リストは、分散して作成されることを特徴とする請求項9記載の編集リスト作成方法。

【請求項16】 上記少なくとも二つの編集リストは、並列して作成されることを特徴とする請求項9記載の編集リスト作成方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、編集リスト作成方法に関し、特に複数の映像及び／又は音声の素材を用いてコンテンツを形成するための編集リスト作成方法に関する。

## 【0002】

【従来の技術】 動画像の編集は、映画フィルムのスプライサー＋セメントに始まったが、ビデオ編集においては、デルマト＋はさみ＋スプライシングテープという黎明期を経て、ダビングによる電子編集時代が長く続いている。

【0003】 ここ十年ほどで普及したノンリニア編集(non-linear editor)も、その延長線上にあるといえる。すなわち、メディアから得た画像を見ながら、主にダビングによって繋ぎ合わせ、必要に応じて画像処理を施し、編集結果を再びメディアに収める、以上一連の作業、あるいはその繰り返しという作業であり、何十年も踏襲されている。

【0004】 しかし、このような伝統手法には以下のような問題点がある。まず、データ圧縮と伸長を繰り返すことが原因での画質劣化が起ることである。アナログで編集を行う時代では、ダビングを繰り返すことにより電子編集でも画質劣化が起きたが、ベースバンド・デジタルの時代に至ってほぼ完璧に解消されたはずであった。しかし、昨今の画像圧縮技術の普及浸透によって、

画質問題は再燃することとなった。データ圧縮と伸長を繰り返すことが原因である。この対策としては、圧縮率を下げたり、圧縮単位を短くする、といった妥協策が講じられるが、メディアのコストや通信時間など、多くの面で不経済であり、しかも画質劣化が完全に解消されるわけでもない。データ圧縮しない場合でも、画像の拡大縮小やピクセルシフト、合成など、いわゆる特殊効果の繰り返し処理によっても、画質劣化と不経済をもたらす。

【0005】次の問題は、従来の編集においては、単一の画像フォーマットにしか対応できなかったことである。NTSCなどの画像フォーマットは、受像機や電波割り当てを主眼に考案・統一された。しかし、昨今HDVSやPC上のウィンドウなど、複数の画像フォーマットが混在で使われるのが、むしろ一般的である。単一のフォーマットしか対応できないことは、素材の収集においても完パケの提供においても大きな弊害となってきた。しかも、とくにコンピュータ上では、ユーザーが勝手な仕様のウィンドウを作れるため、用意されたフォーマットの中から選ぶ方式では対応しきれない。

【0006】次の問題は、従来の編集では、素材を手許に一旦溜めてから使うしかなかったことである。編集時、目的とする画を特定するためタイムコードが使われるが、一巻中の位置を示すに過ぎず、メディアを特定することは基本的に不可能であった。リール番号などでメディアを特定する方法も使われるが、あくまでローカルな運用範囲に留まる。したがって、素材は手許に一旦溜めてから使うしかなく、手間とコストを要する。同様に特殊効果やタイトル挿入なども機種依存で、手許の装置が対応していない加工は不可能なため、編集結果も手許のメディアに記録せざるを得ない。

【0007】以上の問題点の全てを解決するわけではないが、近年、編集リストとしてEDL(Edit Decision List)を作成し、このEDLに記載されたプログラムに従い自動的に電子編集を行うシステムが採用されてきている。

【0008】ところで、前記EDLには、特殊効果など、互換性のない項目も記述されており、システム限定となっている。また、テープの架け替えなどもあり、人手に頼ることが多い。融通が利きにくい点では、完パケを修正したい場合の対応も面倒であり、たとえば最初のコマの前に1コマ挿入するだけでも、全編の再編集もしくはダビングが必要となる。

【0009】このように、現状の編集システムは、現時点でも不経済・不効率な面が多いが、それ以上に、ネットワークのメリットを享受できないという一面がある。現状に固執すれば、時代に取り残された後進的システムになる危険性が高い。それぞれの問題は有機的に絡み合っているため、部分的な対策を施しても新たな問題を引き起こすであろうし、複雑さが連鎖的に増すので、現状

の編集システムのままではいずれは使われなくなることもあり得る。

【0010】

【発明が解決しようとする課題】本発明は、上記実情に鑑みてなされたものであり、データの品質劣化を防ぎながら、またマルチフォーマットのデータに対応し、かつ素材を手許に一旦溜めて使うことなく、さらに処理の互換を可能として、コンテンツを形成することができる編集リストを作成するための編集リスト作成方法の提供を目的とする。

【0011】

【課題を解決するための手段】本発明に係る編集リスト作成方法は、前記課題を解決するために、複数の映像及び／又は音声の素材を用いてコンテンツを形成するための編集リストを作成する編集リスト作成方法において、編集による出力を識別する識別情報とフォーマットを特定するフォーマット宣言文とを少なくとも用いて編集内容を定義する編集定義工程と、入力素材を識別する識別情報と入力素材のフォーマットを特定するフォーマット宣言文とを少なくとも用いて入力素材を定義しながらその入力素材に施す処理を定義していく処理定義工程とを備える。

【0012】このような各工程を備えることにより、本発明に係る編集リスト作成方法は、まず、編集による出力を識別する識別情報とフォーマットを特定するフォーマット宣言文とを少なくとも用いて編集内容を定義し、かつ入力素材を識別する識別情報と入力素材のフォーマットを特定するフォーマット宣言文とを少なくとも用いて入力素材を定義し、さらに入力素材に施す処理を定義していく。

【0013】ここで、上記出力の識別情報及び入力素材の識別情報は、それぞれグローバルユニークな同一体系の識別コードであるので、出力及び入力素材を特定することができる。

【0014】また、入力素材に施す処理方法もグローバルユニークな同一体系の識別コードで記述するので、処理方法も特定することができる。

【0015】

【発明の実施の形態】以下、本発明に係る編集リスト作成方法の実施の形態について図面を参照しながら説明する。この実施の形態は、図1に構成を示すように、本発明の編集リスト作成方法に基づいて編集リストの一種である、編集手続リスト(Edit Procedure List; EPL)を作成するEPL作成部10を内蔵し、コンテンツを形成するコンテンツ形成装置1である。

【0016】コンテンツ形成装置1は、EPL作成部10のほか、EPLに基づいて編集制御信号を生成する編集制御部20と、所定フォーマットのコンテンツを形成するコンテンツ形成部30とを備えて成る。

【0017】このコンテンツ形成装置1には、複数の映

像及び／音声からなる素材が入力される。ここでは、説明の都合上、例えば、ローカルエリアネットワーク（LAN）や、インターネット等のネットワークに接続されたデータベースに格納された素材とする。この素材は、入力端子INを介してEPL作成部10及びコンテンツ形成部30に供給される。

【0018】EPL作成部10は、上記素材をデコーダ11で復号してから、ダウンコンバータ12で例えば間引きし、エンコーダ13でJPEG（Joint Photographic Experts Group；JPEG）方式などの所定の圧縮方式で圧縮して高圧縮のサブ素材にし、サブ素材サーバー14に格納する。サブ素材サーバー14は、ディスクアレイ構成の記録再生部を有したAVサーバーであり、入力する高圧縮サブ素材を順次取り込み、これをファイル化して記録再生部の指定されたアドレス位置に格納する。

【0019】そして、このサブ素材サーバー14に格納された各ファイルの高圧縮サブ素材は、このサブ素材サーバー14に接続された複数台（ここでは3台）の各エディティングターミナル15、16、17に供給され、EPLの作成に用いられる。

【0020】エディティングターミナル15、16、17は、EPL作成部10の主要部をなすものであり、高圧縮サブ素材を読み出して復号し、得られた素材データに基づく映像をモニタに表示する。このとき、オペレータは、エディティングターミナルを介してサブ素材サーバー14を制御し、サブ素材サーバー14に所望の動作（再生、巻き戻し又は早送り等）を実行させながら、モニタに表示された映像を目視確認してEPLを作成する。EPLは、編集に使用される素材を特定するための識別情報と、少なくとも一部の素材のフォーマットを定義するフォーマット宣言文とを有する。

【0021】以下、EPL作成部10で作成されるEPLについて説明する。EPLには、編集に使用される素材を特定するための識別情報を記述する。素材の識別情報は、グローバルユニークな識別コードであり、マテリアル識別コード（Material ID；UMID）と称される。また、このEPLには、上記識別コード体系に準拠して、素材の処理方法を識別する識別コード、すなわち画像処理方法を表す識別コードが定められる。この素材の処理方法を識別する識別コードもグローバルユニークな識別コードであり、プロセス（Process）ID又はユニークtransform ID（UTID）と称される。UTIDにより、フェード、ミックス、スーパー等の処理を識別できる。また、このEPLには、出力に関するメタデータも記述する。これはEPLによる編集結果のIDであり、上記UMIDも含む。また、このEPLには例えば編集の開始点や終了点を時間で示す、タイムコード（TimeCode；TC）も記述される。

【0022】さらに、EPLには、少なくとも一部の素

材のフォーマットを定義するフォーマット宣言文を記述する。このフォーマット宣言文は、入力映像素材の画像フォーマット及び／又は編集出力の画像フォーマットを含むものである。このため、コンテンツ形成装置1により形成されたコンテンツは、上記フォーマット宣言文により宣言されたフォーマットとされる。フォーマット宣言文によりフォーマットを宣言すれば、どのようなフォーマットのコンテンツも形成することができる。

【0023】一つのEPLファイルで記述される時間の範囲は、最小がコマ、最大は無限である。記述の順番は処理に従い、また、複数の処理を同時に実行させるように記述できる。これは並列処理を記述できる“verilog”言語に似るので、それをベースに考えることができる。ただし、verilogにはlinkやglobal, structの概念がないので、流用は不可能である。

【0024】EPLは基本的に変数を持たない。四則演算や関数文はなく命令文（処理と制御）のみからなる。画像データの受渡しは構文中に明記しない。上の行の出力が下に入力されるだけとなる。

【0025】以下に、EPLの具体例を説明する。外部のSD画（525i画）を白黒にし、回想シーンとして同封の画に挿入し、タイトルを追加するという編集を行うためのものである。図2及び図3にはEPLの記述例を示す。図2と図3は、同一のEPL記述例を示すものであり、図2は簡略化したもの、図3は詳細なものである。図4にはEPLの実行のイメージ図を示す。図5には編集後にできる映像を模式的に示す。

【0026】先ず、図2及び図3のEPL記述例について説明する。1行目の下線部（1-1）の「epl」は、後に続く（）内に、EPLの単位を示すことを表すものであり、出力する画像を定義している。同行の下線部（1-2）、すなわち「umid」は、上記出力画像の識別情報であり、グローバルユニークな識別コードUMIDである。具体的には、図3に示すように、「02232203\_824504F1\_0800468A92000201」のように、長いコードであり、まさしくグローバルユニークなものである。同行の下線部（1-3）及び（1-4）は、編集の開始点及び終了点を示すものであり、ここでは開始フレームのタイムコード及び終了フレームのタイムコードが記述されている。開始点が「00:00:00.00」であり、終了点が「00:01:00.02」であることを特定している。同行の下線部（1-5）は、スキャンフォーマットscan\_formatであり、上述した、フォーマット宣言文の一種である。ここでは出力のビデオフォーマット（video format）を示す。「#1080i」は、HD画像のビデオフォーマットである。同行の下線部（1-6）、すなわち「{ }」は、23行目の下線部（23-1）の「{ }」と一緒に、{ }で囲んだ範囲内に画像処理を羅列する、ということを示す。ここでの画像処理は信号の流れに従う。

【0027】2行目の下線部（2-1）、すなわち「so

「source」は、後に続く()内に入力素材を指定するという指定処理を意味する。()内の同行下線部(2-2)の「umid」は画像IDであり、具体的には図3に示す「22083102\_95440497\_0800468A62000232」のように長いコードである。つまり、グローバルユニークな上記UMIDを用いて素材を特定している。同行の下線部(2-3)は開始フレームのタイムコードを示す。図2のように省略されているときは、先頭からということを示す。同行下線部(2-4)の「#embedded.mpg」は素材の取得先、すなわち素材が保管されている場所を示している。同行下線部(2-5)の「#1080i」はスキャンフォーマットscan\_formatであり、上述した、フォーマット宣言文の一種である。ここでは、入力素材のビデオフォーマットを示している。よって入力素材の画像フォーマットを特定することができる。この2行目の下線部(2-1)～(2-5)の記述により、入力素材を、ネットワークに接続された、図4の素材記憶部40から読み出すことができる。

【0028】次に、図2の3行目の下線部(3-1)の「fork」は、22行目の下線部(22-1)の「join」と対になり、入力1に対して複数の並列処理され、出力はjoinで一括加算されることを表す。図4では、符合41と加算器42で挟まれた範囲で行われる処理であり、素材記憶部40からの入力1に対して、複数の並列処理され、加算器42で一括加算されることを示す。その処理の内容は4行目、5行目に示すとおりである。

【0029】4行目の下線部(4-1)の「@」は、後に続く()内の数字により、切り抜き位置を入力側の時間軸で指定していることを示す。そして、()内の同行下線部(4-2)には、開始点が示される。ここでは、入力側の時間軸による「02:10:00.00」が示されている。同行下線部(4-3)の「put」は、後に続く()内に、上記入力素材から切り抜いた部分の貼り付け位置を出力側の時間軸で指定する。ここでは、()内の同行下線部(4-4)及び(4-5)に示すように、出力側の「00:05.00」から「00:25.00」に貼り付けることを示す。これは、入力側の時間軸で指定した切り抜き位置からの入力素材43(図4に示す)を本編カット1として出力側の「00:05.00」から「00:25.00」に貼り付ける処理になる。

【0030】5行目の下線部(5-1)の「@」も、後に続く()内の数字により、切り抜き位置を入力側の時間軸で指定していることを示す。そして、()内の同行下線部(5-2)には、入力側の時間軸による「02:10:20.00」が示される。同行下線部(5-3)の「put」も、後に続く()内に、上記入力素材から切り抜いた部分の貼り付け位置を出力側の時間軸で指定することを示す。ここでは、下線部(5-4)及び(5-5)に示すように、出力側の「00:35.00」から「01:00.02」に貼り付けることを示す。これは、入力側の時間軸で指定した

切り抜き位置からの入力素材44(図4に示す)を本編カット2として出力側の「00:35.00」から「01:00.02」に貼り付ける処理になる。

【0031】次に、6行目の下線部(6-1)の「{」は、11行目の下線部(11-1)の「}」と対をなし、囲んだ範囲内では、入出力1対1で複数の処理を上から順次実行するというを示す。図4に示す処理45に相当する。

【0032】次に、7行目の下線部(7-1)の「source」により、後に続く()内に入力素材を指定する。()内の同行下線部(7-2)の「umid」は、図3に示す「22083102\_95440497\_0800468A62000232」のように長いコードである。同行の下線部(7-3)の「06:54:32.10」は開始フレームのタイムコードを示す。同行下線部(7-4)の「url」は素材の取得先、すなわち素材が保管されている場所を示している。具体的には、図3に示すように、「ftp://43.14.61.62/source/」である。同行下線部(7-5)の「#525i」はスキャンフォーマットscan\_formatであり、上述した、フォーマット宣言文の一種である。入力素材のビデオフォーマットを示している。よって入力素材の画像フォーマットを指定することができる。この7行目の下線部(7-1)～(7-5)の記述により、図4に示した処理45中において、ビデオフォーマットが「#525i」の入力素材を、ネットワーク46を介して読み出し、図4の素材記憶部47に格納する処理を特定できる。同行下線部(7-6)は、このソースが何のシーンに使われるのかを示す。ここでは、「//recollect」であり、回想シーンであることを特定する。

【0033】次に、8行目の下線部(8-1)の「transform」は、後に続く()内にて画像処理を指定することを示す。同行下線部(8-2)の「utid」は、処理IDであり、グローバルユニークな画像処理コード(Transform ID; UTID)が記述される。具体的には、図3に示すように、「1200000000090001」である。同行下線部(8-3)の「url」は、処理方法の取得先、すなわち処理方法が保管されている場所を示している。具体的には、図3に示すように、「ftp://43.14.61.62/exec/」というように記述される。同行下線部(8-4)の「upconv525to1080」は、処理の通称であり、取得先から得られなかったとき、代替え処理を可能にする。ここでは、ビデオフォーマットを525ラインから1080ラインにアップコンバージョンする処理を表す。同行下線部(8-5)の「#1080i」はスキャンフォーマットscan\_formatであり、上述した、フォーマット宣言文の一種である。出力のビデオフォーマットを示している。よって出力の画像フォーマットを指定することができる。入力ビデオフォーマットと同じなら省略することが可能である。この8行目の下線部(8-1)～(8-5)の記述により、図4に示した処理45中において、

図4の素材記憶部47に格納したビデオフォーマットが「#525i」の入力素材を、ビデオフォーマット「#1080i」にアップコンバージョン部48でアップコンバージョンする処理を特定できる。

【0034】次に、9行目の下線部(9-1)の「transform」で()内に画像処理を指定する。同行下線部(9-2)の「utid」の具体例としては、図3に示すように、「120000000001F0001」がある。同行下線部(9-3)の「url」にて、処理方法が保管されている場所を示している。具体的には、図3に示すように、「ftp://43.14.61.62/exec/」というように記述される。同行下線部(9-4)の「"monochrome"」は、処理の通称である。ここでは、1080ラインにアップコンバージョンされた画像を白黒にする処理を表す。なお、ここでは、前行下線部(8-5)にあった「#1080i」の記述を省略している。入力と同じなら省略可である。この9行目の下線部(9-1)～(9-4)の記述により、図4に示した処理45中において、アップコンバージョン部48でビデオフォーマット「#1080i」にアップコンバージョンした画像は、モノクロ処理部49にて白黒画像とされる。

【0035】次に、10行目の下線部(10-1)の「put」にて、後に続く()内に白黒画像の貼り付け位置を出力側の時間軸で指定する。()内の同行下線部(10-2)及び(10-3)にて、出力側の「00:25.00」から「00:35.00」に、貼り付ける処理を特定している。図5に示す回想シーンの貼り付けである。

【0036】次に、12行目の下線部(12-1)の「{」は、21行目の下線部(21-1)の「}」と対をなし、囲んだ範囲内では、入出力1対1で複数の処理を上から順次実行するというを示す。図4に示す処理51に相当する。

【0037】次に、13行目の下線部(13-1)の「source」により、後に続く()内に入力素材を指定する。()内の同行下線部(13-2)のumidは、図3に示す「00000000\_8244041D\_0800468A940000522」のように長いコードである。同行の下線部(13-3)の「00:00:00.00」は開始フレームのタイムコードを示す。同行下線部(13-4)の「url」は画像の取得先であり、具体的には、図3に示すように、「ftp://43.14.61.62/source/\*.jpg」である。同行下線部(13-5)の「#1080i」は入力素材のビデオフォーマットを示している。よって入力素材の画像フォーマットを指定することができる。この13行目の下線部(13-1)～(13-5)の記述により、図4に示した処理51中において、ビデオフォーマットが「#1080i」の入力素材を、ネットワーク52を介して読み出し、図4の素材記憶部53に格納する処理を特定できる。同行下線部(13-6)は、このソースが何のシーンに使われるのかを示す。ここでは、「//title」であり、タイトルに使われ

ることを特定する。

【0038】次に、14行目の下線部(14-1)の「fork」は、20行目の下線部(20-1)の「join」と対になり、入力1に対して複数の並列処理され、出力はjoinで一括加算されることを表す。図4では、素材記憶部53から読み出された素材が2系統で処理され、加算器57で加算されるまでである。2系統での処理内容は、以下のとおりである。

【0039】まず、15行目の下線部(15-1)の「put」にて、後に続く()内に、素材記憶部53から読み出されたタイトル素材54の貼り付け位置を出力側の時間軸で指定する。()内の同行下線部(15-2)及び(15-3)にて、出力側の「00:00.00」から「00:04.00」に、貼り付ける処理を特定している。

【0040】次に、16行目の下線部(16-1)の「{」は、19行目の下線部(19-1)の「}」と対をなし、囲んだ範囲内では、入出力1対1で複数の処理を上から順次実行するというを示す。図4に示す処理51の上半分の処理に相当する。

【0041】17行目の下線部(17-1)の「transform」は、後に続く()内にて画像処理を指定する。同行下線部(17-2)の「utid」は、具体的には、図3に示すように、「1200000000080001」である。同行下線部(17-3)の「url」は、具体的には、図3に示すように、「ftp://43.14.61.62/exec/」である。同行下線部(17-4)の「"fadeout"」は、フェードアウト処理を表す。同行下線部(17-5)では、出力のビデオフォーマットを省略している。入力のビデオフォーマットと同様であるため省略できる。同行下線部(17-6)の「1.0」は固有の因数であり、その処理が必要とする固有の設定値を列挙したものである。17行目の下線部(17-1)～(17-6)の記述により、図4に示した処理51中において、素材記憶部53から供給された「#1080i」の入力素材を、フェードアウト処理部55でフェードアウトする処理を特定できる。

【0042】次に、18行目の下線部(18-1)の「put」にて、後に続く()内に、フェードアウト処理部55からの処理出力56の貼り付け位置を出力側の時間軸で指定する。()内の同行下線部(18-2)及び(18-3)にて、出力側の「00:04.00」から「00:05.00」に、貼り付ける処理を特定している。

【0043】そして、加算器57にてタイトル素材54+フェードアウト処理出力56を加算し、その加算出力を加算器42に供給する。加算器42は、図5に示すように、フェードアウトするタイトルに続いて本編カット1を入れ、白黒の回想シーンを挟んで、本編カット2を入れた、編集結果を出力する。

【0044】次に、図3の下段の構文について説明する。下線部(25-1)の「format」は、後に続く{}にて画像フォーマットを指定する。下線部(25-2)の



「1001/60000」や、下線部(25-3)の「Capturing interval (long/long) [s]」(パラメータ)、或いは下線部(26-1)の「1080/59.94i」、下線部(26-2)の「// Name of format (char)」によって画像フォーマットを指定している。

【0045】また、下線部(27-1)の「format」も、後に続く{}にて画像フォーマットを指定する。下線部(27-2)の「1001/60000」や、下線部(27-3)の「Capturing interval (long/long) [s]」(パラメータ)、或いは下線部(28-1)の「480/59.94i」、下線部(28-2)の「// Name of format (char)」によって画像フォーマットを指定しているのは上記と同様である。

【0046】そして、図3のEPLは、下線部(29-1)の「image」にて同封の画像データの詳細を規定する。()内の下線部(29-2)には、バイナリ部分のバイト数を示す。ここでは、(bit+7)\*8/8である。下線部(29-3)は、ファイル内での名称であり、ここでは「embedded」としている。下線部(29-4)の「binary data...」は、データ本体を特定している。

【0047】以上により、EPLの具体例を説明した。EPLは通常、文字列のままインタープリター(interpreter)で処理される。編集等も文字列のままで行われる。コンパイル(Compile)等する際は、その機材固有のコードとなるので、本書式には定義しない。保存等で圧縮する際は、通常の無損失圧縮(zip, lha の huffman など)を用いる。

【0048】また、UMIDはSMPTE330M basic UMID 32Bytesである。ユニバーサルレベル(Universal label)やインスタンス(Instance)などを省き、マテリアルナンバー(Material number)16Bytesだけでもよい。出力画像は実行処理で発生するが、EPL生成で素材発生と見做し、そこで付加する。

【0049】表記は16進法(hexadecimal)で、目視のため任意にアンダースコア(underscore '\_')を挿入できる。括弧()で括った場合、素材そのものでなく、加工した後の画像を示す。取得先(source や transform の引数)は、文字列の先頭で在り処を区別する。UMIDが予めデータベース登録してあれば取得先は省略可能となる。

【0050】また、タイムコードTCは、一般的な時:分:秒.Frameで、不要な上位桁は省くことができる。入力側はIN点TCのみを記述し、出力側はIN/OUT両方のTCを記述する。出力側は、put()で挿入位置を記述するが、省略するとepl()で示す全編に有効となる。入力側は、source()中の引数、もしくは@()を用いて切り出し位置を記述する。カット編集をする場合は、当然必ず入出力両方のTCを記述する必要がある。入力側のTCは、エフェクト類のみなど、指定する必要のない場合、省略できる。

【0051】上から下の行に受け渡す画像データは、R/G/B/tl か Y/Pb/Pr/tl である。たとえば、Y色差 4:2:2 なら左上画素から Y tl Pb Pr Y tl の繰り返しとなる。また、format{} 指定の総ビット幅より細い経路を通す際は、LSB first で詰め込む。

【0052】透過率tl(translucent)はY毎に付き、通常画像では値が0(不透明)である。このtlは中間的に用いるだけで、通常のepl()実行出力には載せない。

10 【0053】入力画像のデータ形式はファイルの拡張子で判別する。省略可能である。例えば、eplはEPL fileを示す。また、eivはEPLL Intermediate Videoを示す。また、ybrはY/Pb/Prを示す。また、rgbは、Red/Green/Blueを示す。またjpgはJPGw p示す。また、mpgはMPEG1を示す。

【0054】このように入力画像のデータ形式は、各種あるが、標準的にサポートするのは上記とMPEG2全プロフィールである。標準以外の圧縮は、source()直後にtransform()で伸長処理を記述する。上記eivは、tlまで含む非圧縮にformat{} header が付いたものである。このeivは、plug()をoptimizeで解消しないまま実行すると発生する。通常はサーバーがUMID基準で管理しており、EPL上にはファイル名は記述しない。したがって、拡張子を記述するには、wild card '\*'を用いる。

20 【0055】なお、出力画像はモニターや放送用のエンコーダ(encode)にだけ供給されるので、非圧縮のみである。ただし、epl(){}の最後の行にtransform()を記述して、圧縮を定義できる。

【0056】次に、EPLに載せるビデオスキャンフォーマット(video scan format)の書式について説明する。

【0057】将来性や汎用性を考慮し、詳細仕様をいちいち記述する。つまり、通称の"NTSC"と同時に、720x480, 60/1.001, colorimetry,...を列挙する。

【0058】ただし、将来と言っても、あくまでラストで、ピクセルとコマという概念は固持する。つまり本発明のビデオフォーマットは、ラスト映像フォーマットの一般化表現である。

40 【0059】EPL上にformat{}文で載せる際は、以下の例に示す如く、ASCII文字列で記述する。映像信号に載せる際は、指定データ型のbinary 128bytesを、クリップ毎に入れる。

・ Capturing interval ... 撮影時の1コマの時間を示す。long/long [s]  
 ・ Shutter speed ... 撮影時のシャッタースピードを示す。float [s]  
 ・ Shutter timing ... シャッターを開けた時刻を示す。float [s]  
 ・ Display interval ... 表示時の1コマの時間を示す。long/long [s]

- ・ RGB on chromaticity … 三原色RGBと白の色度図上の座標を示す。fixed x 2x 4
- ・ Matrix coefficients … Y/Pb/Pr への変換マトリクスを示す。fixed x 9
- ・ Gamma code … ガンマ変換の記号を示す。char
- ・ Interlace code … インターレースの記号を示す。char
- ・ padding … バイト数あわせ。char x 4
- ・ Picture size … 縦横の有効画素数を示す。short\*short
- ・ Active size … 黒ベタを除いた範囲を示す。short\*short
- ・ Clean aperture … 画質を保証する範囲を示す。short\*short
- ・ Pixel aspect ratio … 各ピクセルの縦横比を示す。float
- ・ Luminance offset … 輝度を開始する位置を示す。float x 2
- ・ Chrominance offset … 色差を開始する位置を示す。float x 2
- ・ Luminance pitch … 輝度の画素の間引きを示す。char x 2
- ・ Chrominance pitch … 色差の画素の間引きを示す。char x 2
- ・ Bit width … Y/Pb/Pr/tl のデータビット幅を示す。char x 4
- ・ Y range … 輝度 100% 白と 0% 黒の値を示す。short x 2
- ・ Other range … 輝度以外の最大値と最小値を示す。short x 2
- ・ Name of format … フォーマットの通称を示す。char x 16

ここで、longは4バイトbytes, shortは2バイトbytes, charは1バイトbyteのアンサインインテジャーunsigned integerである。fixedは2bytes固定小数点で、0x7fff=1と見做す。0x0001 $\approx$ 0.00003になる。また、floatは値として全項目で正のみだが、通常の4bytes IBM浮動小数点となる。

【0060】透過率tlは中間処理に用いるだけで、通常のepl()実行出力信号には載せない。ガンマは、ほかと同様に係数(0.018, 4.5, 1.099, 0.45, 0.099)での表現も可能だが、実運用は変換テーブルである。色差は、2's comp, オフセットバイナリ(offset binary)のいずれも可とし、レンジrangeの大小で表現する。

【0061】図6には具体例を示す。これは、(1080/59.95i)に関するものである。詳細な説明は省略する。

【0062】ここで、図1に戻る。エディティングターミナル15, 16, 17で作成されたEPLは、EPL記憶部18に格納される。このEPL記憶部18に格納されたEPLは、編集制御部20に読み出される。編集

制御部20は、EPLに基づいて編集制御信号を生成し、編集制御信号をコンテンツ形成部30に供給する。

【0063】コンテンツ形成部30は、編集制御部20から供給された編集制御信号に基づいて素材から所定フォーマットのコンテンツを形成する。コンテンツ形成部30には、上述したように入力端子INを介して入力素材が供給される。

【0064】コンテンツ形成部30は、上記入力素材を原素材サーバー31に格納しておく。原素材サーバー31は、ディスクアレイ構成の記録再生部と、AVサーバーであり、供給される各原素材のなかから指定された複数系統の映像音声データを同時にに取り込み、これをそれぞれファイル化して記録再生部内の指定されたアドレス位置に格納する。

【0065】原素材サーバー31は、編集制御部20から供給される編集制御信号に基づいて、格納された各ファイルの原素材を読み出し、原素材サーバー31に接続された編集実行部32に供給する。

【0066】編集実行部32は、複数台(ここでは2台)の各デコーダ33, 34と、この編集実行部32の主要部であるスイッチャー&エフェクター35と、エンコーダ36からなる。各デコーダ33, 34で復号された原素材は、スイッチャー&エフェクター35にて、編集制御部20から供給される編集制御信号に基づいて、編集処理が施された後、エンコーダ36に供給される。エンコーダ36は、編集制御部20から供給される編集制御信号に基づいたフォーマットで、編集処理が施された編集データにエンコード処理を施してから出力端子OUTに供給する。

【0067】編集実行部32内において、スイッチャー&エフェクター35は、デコーダ33及び34から2つの原素材出力が同時に与えられると、A/Bロール編集、すなわち2つの信号を使用した切換編集や挿入編集、或いは重ね合わせ編集等を行う際、その入力される二つの原素材出力を使用してリアルタイムで編集処理を行うようになされている。また、スイッチャー&エフェクター35は、デコーダ33又は34から一つの原素材出力が与えられると、A/Bロール編集を行う際、必要な二つの原素材を時間的に順番に読み出し、先に入力された原素材を一旦内部のバッファに蓄積しておき、続いて入力される原素材と先に蓄積した原素材とを使用してノンリアルタイムで編集処理を行うようになされている。

【0068】以上に説明したように、図1に示したコンテンツ形成装置1は、編集に使用される素材や出力を特定するためのUMIDや、処理方法を特定するためのUTID、さらには入力素材、出力のフォーマットを特定するスキャンフォーマットを、EPLに記述している。このため、編集実行時には、素材をIDで取得できるので、重い画像データを伝送しない運用が可能となる。ま

た、素材をその都度取得させることで、著作権保護や課金が可能となる。また、処理方法がIDで引用できるので、編集環境（機材の能力）を選ばない。また、処理方法そのものの著作権保護や課金も可能となる。また、入力素材を自由に選べる。また、複数フォーマットの出力を同時に得られる。また、EPLを作成する作業は、低画質で進められるので、場所や環境を選ぶことがない。

【0069】このコンテンツ形成装置1の内部にあってEPL作成部10は、本発明の編集リスト作成方法にしたがってEPLを作成しているので、画像データの品質劣化を防ぎながら、またマルチフォーマットの画像データに対応し、かつ素材を手許に一旦溜めて使うことなく、さらに処理の互換を可能として、コンテンツを形成するための編集リストを作成している。

【0070】次に、EPLを数カ所に存在する複数のグループにより分散して作成し、コンテンツを編集するコンテンツ形成システムについて説明する。EPLを用いた編集作業では、素材を直接加工しないので、数カ所に存在する複数のグループで分散して作業を進めることが可能となる。

【0071】図7にコンテンツ形成システム60の全体構成図を示す。素材編集グループ66、編集結果保存・検索グループ74、編集結果利用グループ82がそれぞれLANを構成し、それらLANを接続して第1の編集グループ部を形成している。また、コンピュータグラフィック処理グループ93、クローズドキャプション処理グループ102、地方局111もそれぞれLANを構成し、それらLANを接続して第2の編集グループ部を形成している。

【0072】このシステムでは、ビデオカメラ61で記録された映像及び／又は音声からなる素材ファイル64が、例えばインターネットのような外部ネットワーク63を介して、第1の編集グループ部の素材編集グループ66に供給される。

【0073】まず、第1の編集グループ部を構成する、素材編集グループ66について説明する。素材編集グループ66は、ゲート65とゲート89及び73の中にあり、二つのコンピュータ端末67及び68と、素材サーバー(O)69と、デコーダ・エンコーダ(D)70と、オフライン編集用のイントラフレームサーバー(I)71と、EPLサーバー(E)72とからなる。イントラフレームサーバーは、上記図1にて示したサブ素材サーバー14に相当する。

【0074】素材編集グループ66のオペレータは、コンピュータ端末67又は68を用いて、素材ファイル64を外部ネットワーク63を介して保管場所から或いはビデオカメラ61から、ダウンロードし、素材サーバー69及びイントラフレームサーバー71に格納する。

【0075】イントラフレームサーバー71には、前述したように高圧縮のサブ素材として格納される。すなわ

ち、前記素材ファイル64に収納されている素材をデコーダ・エンコーダ70で復号してから間引き処理等を施し、さらにエンコードして高圧縮のサブ素材にし、このサブ素材をイントラフレームサーバー71に格納する。

【0076】上記二つのコンピュータ端末67及び68は、上記図1に示したエディティングターミナルとしても機能し、イントラフレームサーバー71からサブ素材を読み出して復号し、得られた映像データに基づく映像をモニタに表示する。このとき、オペレータは、コンピュータ端末を介してイントラフレームサーバー71を制御し、イントラフレームサーバー71に所望の動作（再生、巻き戻し又は早送り等）を実行させながら、モニタに表示された映像を目視確認してEPLを作成する。EPLは、上述したように、編集に使用される素材や出力を特定するための識別情報と、少なくとも一部の素材のフォーマットを定義するフォーマット宣言文とを有する。ここで、作成されたEPLは、EPLサーバー72に格納される。このEPLサーバー72に格納されたEPLは、第1の編集グループ部を構成する他のグループや、第2の編集グループ部を構成する各グループによって、分業のために読み出される。

【0077】次に、編集結果保存検索グループ74について説明する。この編集結果保存検索グループ74は、ゲート73とゲート81の中にあり、二つのコンピュータ端末75及び76と、素材サーバー(O)77と、デコーダ・エンコーダ(D)78と、オフライン編集用のイントラフレームサーバー(I)79と、EPLサーバー(E)80とからなる。

【0078】この編集結果保存検索グループ74は、上記EPLサーバー72からEPLを取り出してEPLサーバー80に格納し、格納されたEPLに基づいた編集を行い、仮コンテンツを作成し、保存しておく。これによりこのコンテンツ形成システム60は、必要な編集結果を資料として保存し得るように為される。また、EPL及び編集結果を検索できる。

【0079】次に、編集結果利用グループ82について説明する。この編集結果利用グループ82は、ゲート81から送出側にあり、二つのコンピュータ端末83及び84と、素材サーバー(O)85と、EPLサーバー(E)86と、送出用デコーダ・スイッチャー・エフェクター87と、メディア用デコーダ・スイッチャー・エフェクター88とからなる。

【0080】そして、この編集結果利用グループ82は、上記EPLサーバー72からEPLを取り出してEPLサーバー86に格納し、格納されたEPLに基づいたコンテンツを、上記図1の編集制御部20に対応するコンピュータ端末83及び84を用い、素材サーバー85に格納されている素材等を使って作成し、上記図1の編集実行部32に対応する送出用デコーダ・スイッチャー・エフェクター87や、メディア用デコーダ・スイッチ

ャー・エフェクター88から、送出したり、メディアとして配信する。

【0081】第2の編集グループ部について説明する。第2の編集グループ部は、第1の編集グループ部と、ゲート89及び92に接続された外部ネットワーク90を介して接続しており、プログラムフォルダ91の形態でEPLを双方向に伝送することができる。

【0082】第2の編集グループ部を構成する、コンピュータグラフィック処理グループ93、クローズドキャプション処理グループ102、地方局111は、EPLの作成を分散して行うグループである。

【0083】先ず、コンピュータグラフィック処理グループ93について説明する。コンピュータグラフィック処理グループ93は、二つのコンピュータ端末94及び95と、素材サーバー(O)96と、デコーダ・エンコーダ(D)97と、オフライン編集用のイントラフレームサーバー(I)98と、EPLサーバー(E)99と、コンピュータグラフィックサーバー(CG)100とからなる。イントラフレームサーバー98は、上記図1にて示したサブ素材サーバー14に相当する。

【0084】コンピュータグラフィック処理グループ93のオペレータは、素材編集グループ66のEPLサーバー72から、EPLをプログラムフォルダに収納してダウンロードし、EPLサーバー99に格納する。イントラフレームサーバー98には、コンピュータグラフィック処理を行うときに用いる素材が、デコーダ・エンコーダ97により高圧縮の素材とされて格納される。そして、コンピュータ端末94又は95により、コンピュータグラフィックサーバー(CG)100を用いて、イントラフレームサーバー98から読み出したサブ素材にCG処理を施した編集を行いながら、モニタに表示してEPLを作成する。ここで、作成されたEPLはEPLサーバー99に格納されると同時に、プログラムフォルダに収納されて素材編集グループ66のEPLサーバー72に送られる。

【0085】次に、クローズドキャプション処理グループ102について説明する。クローズドキャプション処理グループ102は、二つのコンピュータ端末103及び104と、素材サーバー(O)105と、デコーダ・エンコーダ(D)106と、オフライン編集用のイントラフレームサーバー(I)107と、EPLサーバー(E)108と、クローズドキャプションサーバー(CC)109とからなる。

【0086】クローズドキャプション処理グループ102のオペレータは、素材編集グループ66のEPLサーバー72から、EPLをプログラムフォルダに収納してダウンロードし、EPLサーバー108に格納する。イントラフレームサーバー107には、クローズドキャプション処理を行うときに用いる素材が、デコーダ・エンコーダ105により高圧縮の素材とされて格納される。

そして、コンピュータ端末103又は104により、クローズドキャプションサーバー(CC)109を用いて、イントラフレームサーバー107から読み出したサブ素材にCC処理を施した編集を行いながら、モニタに表示してEPLを作成する。ここで、作成されたEPLは、EPLサーバー108に格納されると同時に、プログラムフォルダに収納されて素材編集グループ66のEPLサーバー72に送られる。

【0087】次に、地方局111について説明する。地方局111は、二つのコンピュータ端末112及び113と、素材サーバー(O)114と、デコーダ・エンコーダ(D)115と、オフライン編集用のイントラフレームサーバー(I)116と、EPLサーバー(E)117と、地方独自情報(DI)サーバー118とからなる。

【0088】地方局111のオペレータは、素材編集グループ66のEPLサーバー72から、EPLをプログラムフォルダに収納してダウンロードし、EPLサーバー117に格納する。イントラフレームサーバー116には、地方局の編集で用いる素材が、デコーダ・エンコーダ115により高圧縮の素材とされて格納される。そして、コンピュータ端末112又は113により、地方独自情報(DI)サーバー118を用いて、イントラフレームサーバー116から読み出したサブ素材に独自の編集を施しながら、モニタに表示してEPLを作成する。ここで、作成されたEPLは、EPLサーバー118に格納されると同時に、プログラムフォルダに収納されて素材編集グループ66のEPLサーバー72に送られる。

【0089】そして、最終的に素材編集グループ66にて、EPLサーバー72内の各EPLを管理し、各EPLを統合・最適化する。編集結果保存・検索グループ74、編集結果利用グループ82では、前記素材編集グループ66にて統合・最適化されたEPLを用いる。

【0090】以上、説明したように、上記図7に示したコンテンツ形成システムは、編集に使用される素材や出力画像を特定するためのUMIDや、処理方法を特定するためのUTID、さらには入力素材、出力画像のフォーマットを特定するスキャンフォーマットを、EPLに記述している。このため、編集実行時には、素材をIDで取得できるので、重い画像データを伝送しない運用が可能となる。また、素材をその都度取得させることで、著作権保護や課金が可能となる。また、処理方法がIDで引用できるので、編集環境(機材の能力)を選ばない。また、処理方法そのものの著作権保護や課金も可能となる。また、入力素材を自由に選べる。また、複数フォーマットの出力を同時に得られる。また、EPLを作成する作業は、低画質で進められるので、場所や環境を選ぶことがない。また、編集作業では、素材を直接加工しないので、分散したグループにより分業体制で同時に

作業が進められる。

【0091】次に、EPL編集作業の分業の具体例について図8～図14を用いて説明する。インタビュー収録生素材を粗編集してEPL(1)を作成したあと、カット編集のEPL(2-a)、モザイク編集のEPL(2-b)、日本語字幕のEPL(2-c)、英訳字幕のEPL(2-d)を並行して作成し、最後にまとめてオブティマイズしたEPL(3)を作成したのち、実行(完パケ映像信号の出力)する、という想定である。なお、この例ではformat{}や表現が似る個々の処理を省略して書いている。

【0092】図8は、EPL編集作業を分業する、コンテンツ形成システム120の機能ブロック図を示す。ビデオカメラ121で撮影した映像及び/又は音声からなる素材を、ロギング部(粗編集部)122で粗編集してEPL(1)を作成する。このEPL(1)を用い、カッティング編集部123、モザイク編集部124、サブタイトル日本語編集部125、サブタイトル英語編集部126が並行して、それぞれの編集を進めて、カット編集のEPL(2-a)、モザイク編集のEPL(2-b)、日本語字幕のEPL(2-c)、英訳字幕のEPL(2-d)を作成する。もちろん、粗編集部122、カッティング編集部123、モザイク編集部124、サブタイトル日本語編集部125及びサブタイトル英語編集部126は、上記図1に示したEPL作成部10と同様の構成、機能を備えている。

【0093】そして、日本語及び英語統合・最適化部127等が各編集部からのEPL(2-a)、(2-b)、(2-c)、(2-d)を統合して最適化し、EPL(3)を生成する。日本語及び英語統合・最適化部127等は、上記図1に示した編集制御部20の機能も備えている。EPL(3)は、実際に編集を行う、編集実行部128に送られる。

【0094】編集実行部128は、上記図1に示したコンテンツ形成部30と同様の、構成機能を備えてなり、ビデオカメラ121からの素材を内部サーバーに蓄えておいてあるので、上記EPL(3)を用いて編集を実行できる。

【0095】図9は、粗編集部122にて作成された粗編集EPL(1)である。UMID「12345608\_23450467\_0800468A88000021」にて、出力を指定している。そして、編集の開始点及び終了点を、開始フレームのタイムコード及び終了フレームのタイムコードで記述する。ここでは、開始点が「00:00:00.00」であり、終了点が「00:01:03.12」である。この出力のビデオフォーマットは「#1080i」という記述で特定することができる。

【0096】次に、UMID「29543202\_234504C0\_0800468A72000098」により入力素材を識別し、保管場所「file://43.14.61.62/source/」からビデオフォーマット「#1080i」の入力素材を読み出すことを特定している。

【0097】次に、「fork」と「join」の対により、入力1に対して5つの並列処理が行われ、出力はjoinで一括加算されることを表す。先ず第1の処理は、入力側の時間軸「10:00:00.12」からのインタビューワーと一緒に素材を、出力側の時間軸「00:00.00」から「00:03.00」までに貼り付けるものである。第2の処理は、入力側の時間軸「10:01:05.23」からの話しているときの素材を、出力側の時間軸「00:03.00」から「00:20.13」までに貼り付けるものである。第3の処理は、入力側の時間軸「10:02:54.11」からの話しているときの素材を、出力側の時間軸「00:20.13」から「00:40.09」までに貼り付けるものである。第4の処理は、入力側の時間軸「10:05:18.19」からの話し手の頭の背景画像を、出力側の時間軸「00:40.09」から「00:43.10」までに貼り付けるものである。第5の処理は、入力側の時間軸「10:07:33.03」からの話しているときの素材を、出力側の時間軸「00:43.10」から「01:03.12」までに貼り付けるものである。

【0098】図10は、カッティング編集部123にて作成されたカット編集用のEPL(2-a)である。UMID「00521209\_234504A3\_0800468A89000128」にて、出力を特定し、編集の開始点及び終了点を、「00:00:00.00」及び「00:00:45.00」と特定している。また、出力のビデオフォーマットも「#1080i」という記述で特定することができる。

【0099】次に、UMID「12345608\_23450467\_0800468A88000021」で、ビデオフォーマット「#1080i」の入力素材を、保管場所「file://43.14.61.62/source/」から取り出すことを特定している。

【0100】そして、上記入力素材を用いて、以下の5つのカット編集を行う。先ず第1の処理は、入力側の時間軸「00:00:00.07」からの素材を、出力側の時間軸「00:00:00.00」から「00:23.04」までに貼り付ける。第2の処理は、入力側の時間軸「00:00:32.23」からの素材を、出力側の時間軸「00:23.04」から「00:28.17」までに貼り付ける。第3の処理は、入力側の時間軸「00:00:40.09」からの素材を、出力側の時間軸「00:28.17」から「00:31.18」までに貼り付ける。第4の処理は、入力側の時間軸「00:00:44.12」からの素材を、出力側の時間軸「00:31.18」から「00:37.29」までに貼り付ける。第5の処理は、入力側の時間軸「00:00:52.21」からの素材を、出力側の時間軸「00:37.29」から「00:45.00」までに貼り付ける。これら5つのカット編集は一括加算される。

【0101】上記5つの処理を括った「fork」と「join」という記述の後に、「jack(mosaic)」及び「jack(super)」という記述がある。「jack」は、統合・最適化部127にてオブティマイズ時に「child EPL」からの受け取り位置を示すものである。(mosaic)及び(super)は、ユーザ任意で与える仮のIDである。ここで

は、モザイク編集部124で行うモザイク編集で作成したEPLを受け取る位置を示すので、(mosaic)を用いた。また、サブタイトル日本語編集部125で行う日本語字幕編集で作成したEPLを受け取る位置を示すので(super)を用いた。オブティマイズした後、「jack (mosaic)」及び「jack (super)」という記述は消える。

【0102】図11は、モザイク編集部124にて作成されたモザイク編集用のEPL(2-b)である。UMID「21341109\_23450411\_0800468A9B000032」にて、出力を特定し、編集の開始点及び終了点を、「00:00:00.00」及び「00:01:03.12」と特定している。また、出力映像のビデオフォーマットも「#1080i」という記述で特定することができる。

【0103】次に、UMID「12345608\_23450467\_0800468A88000021」で、ビデオフォーマット「#1080i」の入力素材を、保管場所「file://43.14.61.62/source/」から取り出すことを特定している。

【0104】そして、上記入力素材を用いて、以下の2つのモザイク編集を行う。先ず第1のモザイク編集は、保管場所「ftp://43.14.61.62/exec/」に保管されている、「12000000000C0004」という処理ID(UTID)で特定された画像処理を使って、入力素材と同様のビデオフォーマット「#1080i」で、「=384x232+880+128」という設定値で行う。このモザイク編集で得られた編集結果は、開始点「00:03.00」から終了点「00:40.09」に貼り付ける。第2のモザイク編集は、保管場所「ftp://43.14.61.62/exec/」に保管されている、「12000000000C0004」という処理ID(UTID)で特定された画像処理を使って、入力素材と同様のビデオフォーマット「#1080i」で、「=400x256+864+96」という設定値で行う。このモザイク編集で得られた編集結果は、開始点「00:43.00」から終了点「01:03.12」に貼り付ける。これら2つのモザイク編集は一括加算される。

【0105】上記2つの処理を括った「fork」と「join」という記述の後に、「plug (mosaic)」という記述がある。「plug」は、統合・最適化部127にてオブティマイズ時に「parent EPL」への受け渡し位置を示すものである。「(mosaic)」は、ユーザ任意で与える仮のIDである。オブティマイズした後、「plug (mosaic)」という記述は消える。

【0106】図12は、サブタイトル日本語編集部125にて作成された日本語字幕編集用のEPL(2-c)である。UMID「12221109\_234504FB\_0800468AC1000341」にて、出力を特定し、編集の開始点及び終了点を、「00:00:00.00」及び「00:01:03.12」と特定している。また、出力のビデオフォーマットも「#1080i」という記述で特定することができる。

【0107】次に、UMID「12345608\_23450467\_0800468A88000021」で、ビデオフォーマット「#1080i」の入力素材を、保管場所「file://43.14.61.62/source/」から

ら取り出すことを特定している。

【0108】そして、上記入力素材を用いて、以下の編集を行う。先ず、保管場所「ftp://43.14.61.62/exec/」に保管されている、「1200000000130081」という処理ID(UTID)で特定された処理方法を使って、入力素材と同様のビデオフォーマット「#1080i」で、「飛行機に乗った男」というサブタイトルを作成する。このサブタイトル作成の結果は、開始点「00:00.00」から終了点「00:10.00」に貼り付ける。次に、保管場所「ftp://43.14.61.62/exec/」に保管されている、「1200000000120081」という処理ID(UTID)で特定された処理方法を使って、入力素材と同様のビデオフォーマット「#1080i」で、「離陸したとたん、気を失った。」という日本語字幕を、開始点「00:01.00」から終了点「00:04.02」にスーパーインポーズ(貼り付ける)する。次に、同じ保管場所に保管されている、同じ処理方法を使って、同じビデオフォーマットで、「目覚めたのは南の島だった、」という日本語字幕を、開始点「00:04.12」から終了点「00:07.29」にスーパーインポーズする。そして、いくつかの日本語字幕を画像に貼り付け、最後に、同じ保管場所に保管されている、同じ処理方法を使って、同じビデオフォーマットで、「いつか夢で見たはずの景色が広がっていた。」という日本語字幕を、開始点「01:00.12」から終了点「01:03.12」にスーパーインポーズする。これらの編集は一括加算される。

【0109】上記処理を括った「fork」と「join」という記述の後に記述している、「plug (super)」は、既に説明したように、統合・最適化部127にてオブティマイズ時に「parent EPL」への受け渡し位置を示すものである。

【0110】図13には、サブタイトル英語編集部126にて作成された英訳字幕編集のEPL(2-d)を示す。このEPLは、上記図12を用いて説明した日本語字幕編集のEPLの英語版である。サブタイトルや翻訳処理のIDが異なっているが、略同のような内容であるので、ここでは説明を省略する。

【0111】そして、図14には、日本語統合・最適化部127が各編集部からのEPL(2-a),(2-b),(2-c)を統合・最適化して作成した、EPL(3)を示す。図10~図12にて説明した、カット編集、モザイク編集、日本語字幕編集を統合・最適化することによって得たEPLである。詳細な説明は省略する。

【0112】このほかに、英語統合・最適化部128でEPL(2-a),(2-b),(2-d)を統合・最適化することにより、同様な英語版が生成できるが、ここでは省略した。

【0113】以上の例では、二段階のカット編集を経ており、仮編集の継ぎ目を本編集で継ぎ直さず、そのまま活かしている箇所があるため、両者を統合したときの編

集点が多くなった。

【0114】また、ここでは、素材もタイトラーもエフェクターも、全て同じサーバーから取得している。しかも、明記していないが、編集実行する装置と同一のLANに置かれている。運用上、安全な方法ではあるが、遠くのサーバーから取得することが多くなると予想される。

【0115】なお、参考までに、統合・最適化を、NTSC用に変換する例を、図15に示す。素材の入力元のビデオフォーマットは、「#1080i」であるが、NTSC用への変換であるので、処理と出力のフォーマットは、「#525i」となる。したがって、これを統合すると、EPL(3)の最後に「down converter」の付いたものが生成される。また、処理順が明らかな場合は、plug(), jack()による接続は不要である。

【0116】このように、図8に構成を示した、コンテンツ形成システム120は、EPL編集作業を分業することができる。編集作業では、素材を直接加工しないので、分業体制で同時に作業を進められるためである。また、各編集部で作成されたEPLには、入力素材、出力映像のフォーマットをスキャンフォーマットで特定している。このため、入力素材は自由に選べる。また、複数フォーマットの出力映像が同時に得られる。

【0117】なお、遠隔地で編集作業をする具体例としては、上記図7に示したシステムの他にも、図16に示すようなコンテンツ形成システム130を構成することができる。ビデオカメラ131で撮影した映像及び／又は音声からなる素材を、低レートエンコーダ132にて圧縮し、サブ素材として携帯電話133又はインターネット134で編集部135に送る。編集部135では、サブ素材を、所望の動作(再生、巻き戻し又は早送り等)で実行させながら、モニタに表示された映像を目視確認してEPLを作成する。EPLは、編集に使用される素材や出力画像を特定するための識別情報と、少なくとも一部の素材のフォーマットを定義するフォーマット宣言文とを有する。そして、生成したEPLを携帯電話136又はインターネット137で、編集実行部140に送る。編集実行部140には、ビデオカメラ131からの素材も、ビデオテープ138のハンドキャリー又はインターネット139により送られる。そして、編集実行部140は、上記素材を内部サーバーに蓄えているので、上記EPLを用いて、編集を実行できる。

【0118】ここまでの説明は、編集に使用される素材や出力を特定するためのUMIDや、処理方法を特定するためのUTID、さらには入力素材、出力映像のフォーマットを特定するスキャンフォーマットを、EPLに記述したことによって課題を解決できたことについて説明した。

【0119】しかし、従来の課題としては、画質の劣化に関するものもあった。例えば、ビデオカメラで収録した映像にいくつかのカット(映像シーン)をつなぎ合わ

せるカット編集と、テロップ文字をスーパーインポーズするテロップ編集を施してから、完成した映像を放送するまでも、データ圧縮と伸張を繰り返すことになり、画質の劣化が生じる。この具体例を、図17の処理遷移図を用いて説明する。まず、ビデオカメラ150で撮影された映像は、ビット・レート・リダクション(Bit Rate Reduction: BRR)エンコード(enc)処理151にて圧縮され、ビデオテープに映像素材152として収録される。この映像素材152に対してカット編集を施すときには、圧縮されている映像素材を一旦、デコード

10 (dec)処理153により伸長してから、編集処理(Edit1)154にてカット編集する。カット編集を施した映像素材は、再度、エンコード処理155で圧縮されてから映像素材156としてビデオテープに記録される。次に、この映像素材156に対してテロップ編集を施すときには、圧縮されている映像素材をまた、デコード(dec)処理157により伸長してから、編集処理(Edit2)158にてテロップをスーパーインポーズする。テロップ編集を施した映像素材は、再々度、エンコード処理159で圧縮されてから映像素材160としてビデオテープに記録される。そして、この映像素材160は、放送日に合わせて、放送用のNTSCフォーマットに変換されるため、一旦、デコード処理161されてから、NTSCエンコード処理152され、例えば地上波用のアンテナ163からテレビジョン放送波として送出される。この一連の処理において、映像素材に対して4回の圧縮処理が施されている。伸長処理は3回である。ここに画質劣化が生じることとなる。

【0120】この画質劣化という問題を解決するためにも、前記コンテンツ形成装置は有効である。図18には、その具体例を示す。この具体例は、ビデオカメラ170で撮影された映像及び／又は音声からなる素材を、ビット・レート・リダクション(Bit Rate Reduction: BRR)エンコード(enc)処理171にて圧縮し、ビデオテープに映像素材172として収録する。この映像素材172は、EPL作成処理173に送られる。

【0121】EPL作成処理173は、カット編集とテロップ編集をEPLを生成することによって行う。映像素材172に対してカット編集を施すときには、圧縮されている映像素材を一旦、デコード(dec)処理174により伸長してから、編集処理(Edit1)175にてカット編集を行う。このカット編集は、EPLを作成することによって行われる。また、テロップ編集は、カット編集処理175により作成されたEPLを用いて、テロップ編集処理177により行われる。具体的には、テロップ編集処理177は、デコーダ176でデコードされた素材をモニタしながら、テロップ編集のEPLを作成する。

【0122】EPL作成処理173で作成された、EPLは、編集実行処理179に送られる。編集実行処理1

79には、デコード処理178によりデコードされた素材も供給されている。そして、編集実行処理179は、前記素材に対し、前記EPLにしたがった、カット編集及びテロップ編集を施し、その編集結果をNTSCエンコード処理180に送る。そして、NTSCエンコード処理180により、NTSCフォーマットとされたデータが、例えば地上波用のアンテナ181からテレビジョン放送波として送出される。この一連の処理においては、映像素材に対して2回の圧縮処理が施されただけである。また、デコード処理も実質的には1回だけしか行われてない。

【0123】したがって、この図18に示す、コンテンツ形成システムは、画質の劣化を上記図17に示した従来方式よりも抑えることができる。すなわち、編集途中で圧縮伸長を従来ほど繰り返さないで、画質を保つことができる。

【0124】また、前記コンテンツ形成装置は、例えば、共通番組を、フォーマットを異ならせた2系統で、サイマルキャスト(Simulcast)放送するときにも、画質を保つことができる。図19及び図20を用いてサイマルキャスト放送について説明する。

【0125】図19は、NTSCフォーマットとハイビジョンフォーマットの両方でサイマルキャスト放送をするための、従来のシステムである。このサイマルキャスト放送システム190は、ビデオテープ191から読み出した「1080i」のハイビジョンフォーマットの映像と、ビデオテープ192から読み出した「525i」のNTSCフォーマットの映像をアップコンバージョン部193により「1080i」した映像とを用いて編集処理194する。編集処理194は「1080i」フォーマットで行い、ハイビジョンフォーマットの出力映像を生成する。そして、ハイビジョン用に一方を、アンテナ195から放送波として送出する。他方、NTSC用には、ダウンコンバージョン196で525iにした映像をアンテナ197から放送波として送出する。

【0126】よって、このシステム190では、「525i」のNTSCフォーマットの映像を一旦、「1080i」にアップコンバージョンしたのち、「1080i」で編集し、再度「1080i」出力を「525i」にダウンコンバージョンしてNTSC用としている。NTSC用の映像についても編集処理は「1080i」で行い、放送前に「525i」にしている。

【0127】これに対して、前記コンテンツ形成装置を適用した、図20に示すサイマルキャスト放送システム200によれば、NTSC用の映像の画質を劣化させることがない。すなわち、図示しない、例えば図1に示したEPL作成部10により作成したEPLに基づいて編集実行部203及び編集実行部204が編集を実行したとき、本システム200では、編集後のNTSC用の映像(「525i」)を、アップコンバージョンとダウンコン

バージョンを繰り返すことなく、生成できる。

【0128】このシステム200の編集実行部203又は204で行われる、HD画像(1080i)からSD画像(525i)へ、またSD画像からHD画像へのビデオフォーマット変換について図21を用いて説明する。

【0129】この図21において、入力ビデオ信号は、フィールドメモリ211とフィールドメモリ212に順番に蓄えられる。動き検出部210は、二つのフィールドメモリ211及び212により、時間軸上での差が与えられた、フィールド画像から動きを検出する。この動き検出部210による動き検出の結果、すなわちフレーム画像に動きがあった／フィールド画像に動きがあったことを示す、フレーム／フィールドのtap選択信号は、トランスバーサル型の垂直フィルタ213に供給される。トランスバーサル型の垂直フィルタ213は、二つのフィールドメモリ211及び212によって時間差が与えられた前記フィールド画像に、前記フレーム／フィールドのtap選択信号に基づいて垂直フィルタ処理を施す。前記垂直フィルタ213のフィルタ出力は、トランスバーサル型の水平フィルタ214に供給される。前記水平フィルタ214は、前記垂直フィルタ出力に水平フィルタ処理を施す。この水平フィルタ214の水平フィルタ出力は、イメージエンハンス部215に供給される。

【0130】イメージエンハンス部215は、前記水平フィルタ出力の高域成分を強調する。イメージエンハンス部215からのエンハンス出力は、画像信号変換部216に供給される。画像信号変換部216は、Y/Pb/Pr信号をRGB信号に変換する。このRGB信号は、逆ガンマ補正部217に供給され、逆ガンマ補正がかけられる。この逆ガンマ補正出力は、色彩変換部(カラーリメトリコンバージョン部)218に供給される。カラーリメトリコンバージョン部218は、前記逆ガンマ補正出力に色彩変換処理を施す。この色彩変換処理出力は、ガンマ補正部219に供給される。

【0131】ガンマ補正部219は、前記色彩変換処理出力にガンマ補正処理を施し、画像信号変換部220に供給する。画像信号変換部220は、RGB信号をY/Pb/Pr信号に変換する。このY/Pb/Pr信号が出力ビデオ信号となる。

【0132】したがって、この図21に示した構成において、画質変換処理部216又は画質変換処理部220での変換処理等を換えることによって、ダウンコンバージョン処理又はアップコンバージョン処理に対応することができる。

【0133】

【発明の効果】以上、本発明に係る編集リスト作成方法は、編集による出力を識別する識別情報とフォーマットを特定するフォーマット宣言文とを少なくとも用いて編集内容を定義し、かつ入力素材を識別する識別情報と入



力素材のフォーマットを特定するフォーマット宣言文とを少なくとも用いて入力素材を定義し、さらに入力素材に施す処理を定義していくので、データの品質劣化を防ぎながら、またマルチフォーマットのデータに対応し、かつ素材を手許に一旦溜めて使うことなく、さらに処理の互換を可能として、コンテンツを形成することができ編集リストを作成できる。

【0134】また、編集に使用される素材や編集出力を特定するための識別情報や、処理方法を特定するための識別情報、さらには入力素材、出力映像のフォーマットを特定するフォーマット宣言文を、編集リストに記述するので、編集実行時には、素材をIDで取得できるので、重い画像データを伝送しない運用が可能となる。また、素材をその都度取得させることで、著作権保護や課金が可能となる。また、処理方法がIDで引用できるので、編集環境（機材の能力）を選ばない。また、処理方法そのものの著作権保護や課金も可能となる。また、入力素材を自由に選べる。また、複数フォーマットの出力を同時に得られる。また、EPLを作成する作業は、低画質で進められるので、場所や環境を選ぶことがない。また、編集作業では、素材を直接加工しないので、分散したグループにより分業体制で同時に作業が進められる。

#### 【図面の簡単な説明】

【図1】本発明の実施の形態となる、コンテンツ形成装置の構成を示すブロック図である。

【図2】EPLの記述例を示す図である。

【図3】EPLの記述例を示す図である。

【図4】EPLの実行イメージを示す図である。

【図5】EPLを用いた編集後にできた映像を模式的に示す図である。

【図6】ビデオスキャンフォーマット（video scan format）の具体例を示す図である。

【図7】コンテンツ形成システムの全体構成を示す図である。

【図8】EPLによる編集の分業を行う、コンテンツ形

成システムを示す図である。

【図9】上記図8のコンテンツ形成システムの粗編集部にて作成された粗編集EPL（1）を示す図である。

【図10】上記図8のコンテンツ形成システムのカット編集部にて作成されたカット編集EPL（2-a）を示す図である。

【図11】上記図8のコンテンツ形成システムのモザイク編集部にて作成されたモザイク編集EPL（2-b）を示す図である。

10 【図12】上記図8のコンテンツ形成システムの日本語字幕編集部にて作成された日本語字幕編集EPL（2-c）を示す図である。

【図13】上記図8のコンテンツ形成システムの英語字幕編集部にて作成された英語字幕編集EPL（2-d）を示す図である。

【図14】上記図8のコンテンツ形成システムの日本語統合・最適化部にて作成された最適化EPL（3）を示す図である。

20 【図15】統合・最適化を、NTSC用に変換したEPLを示す図である。

【図16】遠隔地で編集作業をする他の具体例を示す図である。

【図17】収録から放送までに生じる画質劣化という課題を説明するための図である。

【図18】上記画質劣化を解決する、本発明のコンテンツ形成装置及び方法の具体例を示す図である。

【図19】NTSCフォーマットとハイビジョンフォーマットの両方でサイマルキャスト放送を行う、従来のシステムの構成図である。

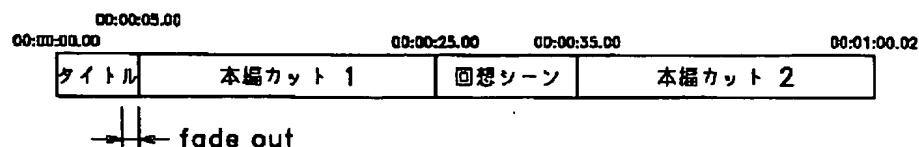
30 【図20】本発明のコンテンツ形成装置及び方法を適用した、サイマルキャスト放送システムの構成図である。

【図21】ビデオフォーマット変換を説明するための図である。

#### 【符号の説明】

1 コンテンツ形成装置、10 EPL作成部、20 編集制御部、30 コンテンツ形成部

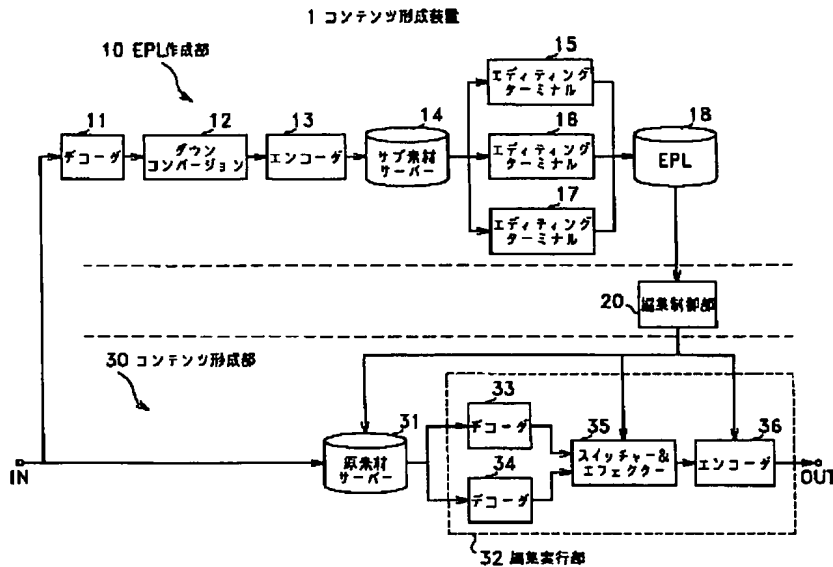
【図5】



【図15】

```
api(20122809_23450489_0800468A89000124, 00:00:00.00, 00:00:45.00, #525f) |
source(09482509_234504A3_0800468A89000126, , filter//43.14.61.62/source/,
#108M);
transform(1200000000090002, ftp://43.14.61.62/exec/,
"downconv1080to525", #525f);
```

【図1】



【図2】

```

1  epl(urlid, 00:00:00.00, 00:01:00.02, #1080i)
2  source(urlid, #embedded.mpg, #1080i);
3  fork
4  @ (02:10:00.00) put(00:05.00, 00:25.00);
5  @ (02:10:20.00) put(00:35.00, 01:00.02);
6  |
7  source(urlid, 06:54:32.10, url, #5250); //recalled
8  transform(urlid, url, "upconv525to1080" #1080i);
9  transform(urlid, url, "monochrome");
10 put(00:25.00, 00:35.00);
11 |
12 |
13 source(urlid, 00:00:00.00, url, #1080i); //file
14 fork
15 put(00:00.00, 00:04.00);
16 |
17 transform(urlid, url, "fadeout", 1.0);
18 put(00:04.00, 00:05.00);
19 |
20 join
21 |
22 join
23 |

```

【図6】

```

format {
1001/60000, // Capturing interval (long/long) [s]
0.0166, 0.0, // Shutter speed and timing (float) [s]
1001/60000, // Display interval (long/long) [s]
0.64, 0.33, // R on chromaticity (fixed)
0.3, 0.6, // G on chromaticity (fixed)
0.15, 0.06, // B on chromaticity (fixed)
0.3127, 0.329, // White on chromaticity (fixed)
0.2126, 0.7152, 0.0722, // Y Matrix coefficient (fixed)
-0.08194, -0.3854, 0.5, // Pb Matrix coefficient (fixed)
0.5, -0.4542, -0.03279, // Pr Matrix coefficient (fixed)
0, // Gamma code (char)
1, // Interlace code (char)
0, 0, 0, 0, // padding
1920*1080, // Picture size (short*short)
1920*1080, // Active size (short*short)
1888*1062, // Clean aperture (short*short)
1.0, // Pixel aspect ratio (float)
0.5, 0.5, // Luminance sampling offset (float)
1.0, 0.5, // Chrominance offset (float)
1, 1, // Luminance pitch (char)
2, 1, // Chrominance pitch (char)
10, 10, 10, 10, // Bit width Y/Pb/Pr/I (char)
940, 64, // Y reference and offset level (short)
960, 64, // Other maximum and minimum level (short)
"1080/59.94i" // Name of format (char)
} 1080i;

```

【図3】

```

epl(02232203_824504F1_0800468A92000201, 00:00:00.00, 00:01:00.02, #1080i) |
(1-2) source(22083102_95440497_0800468A62000232, , embedded.mpg, #1080i);
      fork
        @ (02:10:00.00) put(00:05.00, 00:25.00); (2-2)
        @ (02:10:20.00) put(00:35.00, 01:00.02);
      |
      source(22083102_95440497_0800468A62000232, 06:54:32.10,
(7-4) ftp://43.14.61.62/source/, #525i); //recollect (7-2)
      transform(12000000000090001, ftp://43.14.61.62/exec/,
        "upconv525to1080", #1080i); (8-2) (8-3)
      transform(120000000001F0001, ftp://43.14.61.62/exec/,
        "monochrome"); (9-2) (9-3)
      put(00:25.00, 00:35.00);
      |
      source(00000000_8244041D_0800468A940000522, 00:00:00.00,
      ftp://43.14.61.62/source/*, .jpg, #1080i); //title (13-2)
      fork (13-2)
        put(00:00.00, 00:04.00);
      |
      transform(1200000000080001, ftp://43.14.61.62/exec/,
        "fadeout", , 1.0);
      put(00:04.00, 00:05.00);
      |
      join
    }
  }
  join
}

(25-1) format { 1001/60000, // Capturing Interval (long/long) [s]
  -----< 中略 >----- (25-2) (25-3)
  "1080/59.94i" //Name of format (char)
  { 1080i; (26-1) (26-2)
  } (26-3)
(27-1) format { 1001/60000, // Capturing Interval (long/long) [s]
  -----< 中略 >----- (27-2) (27-3)
  "480/59.94i" //Name of format (char)
  { 525i; (28-1) (28-2)
  } (28-3)
(29-1) image(3888144) embedded;
      :binary data... (29-2) (29-3)
      (29-4)

```

【図9】

--- (1) 組編集 ---

```

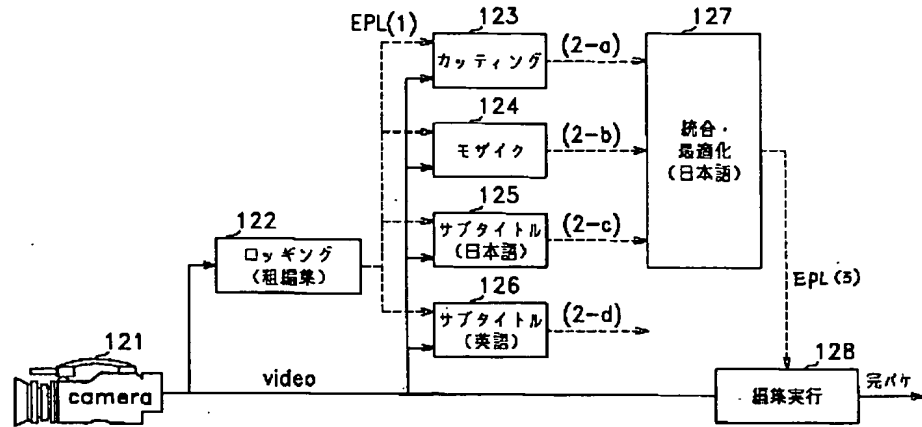
epl(12345608_23450467_0800468A88000021, 00:00:00.00, 00:01:03.12, #1080i) {
  source(29543202_234504C0_0800468A72000098, , file://43.14.61.62/source/,
#1080i);
  fork
    @ (10:00:00.12) put(00:00.00, 00:03.00); // with Interviewer
    @ (10:01:05.23) put(00:03.00, 00:20.13); // telling
    @ (10:02:54.11) put(00:20.13, 00:40.09); // telling
    @ (10:05:18.19) put(00:40.09, 00:43.10); // back of his head
    @ (10:07:33.03) put(00:43.10, 01:03.12); // telling
  join
}

```

【圖 7】

【図8】

## 120 コンテンツ形成システム



【図10】

--- (2-a) カット編集 ---

```
epi(00521209_234504A3_0800468A89000128, 00:00:00.00, 00:00:45.00, #1080i) {
  source(12345608_23450467_0800468A88000021, ,
  file://43.14.61.62/source/, #1080i);
  fork
    @00:00:00.07 put(00:00.00, 00:23.04);
    @00:00:32.23 put(00:23.04, 00:28.17);
    @00:00:40.09 put(00:28.17, 00:31.18);
    @00:00:44.12 put(00:31.18, 00:37.29);
    @00:00:52.21 put(00:37.29, 00:45.00);
  join
  jack(mosaic);
  jack(super);
}
```

【図11】

--- (2-b) モザイク ---

```
epi(21341109_23450411_0800468A98000032, 00:00:00.00, 00:01:03.12, #1080i) {
  source(12345608_23450467_0800468A88000021, 00:00:00.00,
  file://43.14.61.62/source/, #1080i);
  fork
    transform(12000000000C0004, ftp://43.14.61.62/exec/,
    "mosaic", , =384x232+880+128);
    put(00:03.00, 00:40.09);
  }
  transform(12000000000C0004, ftp://43.14.61.62/exec/,
  "mosaic", , =400x256+864+96);
  put(00:43.00, 01:03.12);
  join
  plug(mosaic);
}
```

【図12】

```

--- (2-c) 日本語字幕 ---
epi(12221109_234504FB_080046BAC1000341, 00:00:00.00, 00:01:03.12, #1080f) {
  source( (12345808_23450467_080046BAC10000021), 00:00:00.00,
  file://43.14.61.62/source/, #1080f);
  fork
  |
  transform(1200000000130081, ftp://43.14.61.62/exec/,
  "subtitle", "飛行機に乗った男");
  put(00:00.00, 00:10.00);
  |
  transform(1200000000120081, ftp://43.14.61.62/exec/,
  "translation", "「運命したとたん、男を失った。」");
  put(00:01.02, 00:04.02);
  |
  transform(1200000000120081, ftp://43.14.61.62/exec/,
  "translation", "「目覚めたのは島の奥だった。」");
  put(00:04.12, 00:07.28);
  :
  :
  {
  transform(1200000000120081, ftp://43.14.61.62/exec/,
  "translation", "「いつか夢で見たはずの景色が広がっていた。」");
  put(01:00.12, 01:03.12);
  }
  join
  plug(super);
}

```

【図13】

```

--- (2-d) 英訳字幕 ---
epi(23001109_234504E4_080046BAC1000288, 00:00:00.00, 00:01:03.12, #1080f) {
  source( (12345808_23450467_080046BAC10000021), 00:00:00.00,
  file://43.14.61.62/source/, #1080f);
  fork
  |
  transform(1200000000110001, ftp://43.14.61.62/exec/,
  "subtitle", "A man on the airplane");
  put(00:00.00, 00:10.00);
  |
  transform(1200000000120001, ftp://43.14.61.62/exec/,
  "translation", "The moment the plane took off, I fainted.");
  put(00:01.03, 00:04.00);
  |
  transform(1200000000120001, ftp://43.14.61.62/exec/,
  "translation", "I woke up to find myself on an island in the south.");
  put(00:04.14, 00:07.28);
  :
  :
  {
  transform(1200000000120001, ftp://43.14.61.62/exec/,
  "translation", "I saw a view spreading which I must have dreamt of before.");
  put(01:00.13, 01:03.12);
  }
  join
  plug(super);
}

```

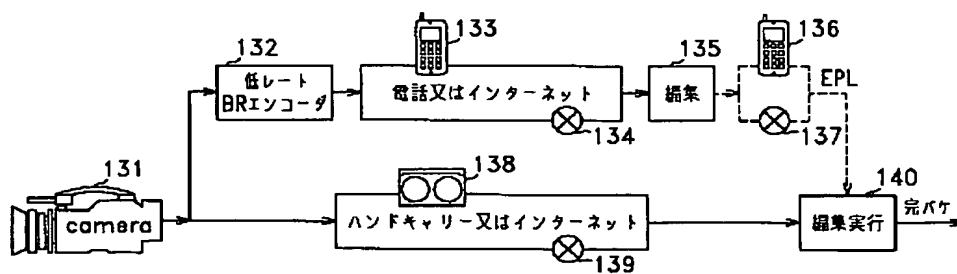
【図14】

```

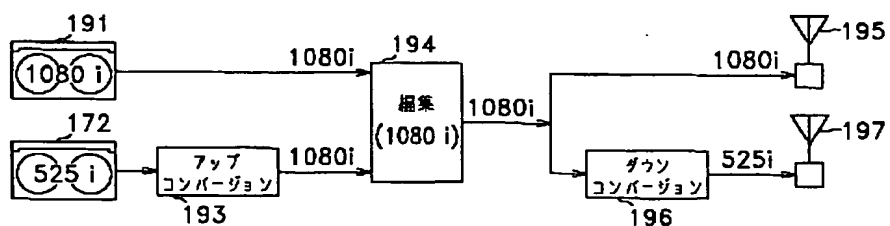
--- (3) 集合・最適化 (日本語版) ---
epi(09482509_234504A3_080046BAC10000128, 00:00:00.00, 00:00:45.00, #1080f) {
  source(29543202_234504C0_080046BAC10000098, file://43.14.61.62/source/,
  #1080f);
  fork
  |
  @ (10:00:00.19) put(00:00.00, 00:02.23);
  @ (10:01:05.23) put(00:02.23, 00:20.06);
  @ (10:02:54.11) put(00:20.06, 00:23.04);
  @ (10:03:06.21) put(00:23.04, 00:28.17);
  @ (10:05:18.19) put(00:28.17, 00:31.18);
  @ (10:07:34.05) put(00:31.18, 00:37.29);
  @ (10:07:42.14) put(00:37.29, 00:43.00);
  join
  fork
  |
  transform(12000000000C0004, ftp://43.14.61.62/exec/,
  "mosaic", "384x232+880+128");
  put(00:02.23, 00:28.17);
  |
  transform(12000000000C0004, ftp://43.14.61.62/exec/,
  "mosaic", "400x256+864+96");
  put(00:31.18, 00:43.00);
  join
  fork
  |
  transform(1200000000130081, ftp://43.14.61.62/exec/,
  "subtitle", "飛行機に乗った男");
  put(00:00.00, 00:09.23);
  :
  :
  {
  transform(1200000000120081, ftp://43.14.61.62/exec/,
  "translation", "「いつか夢で見たはずの景色が広がっていた。」");
  put(00:42.00, 00:45.00);
  }
  join
}

```

## 130 コンテンツ形成システム

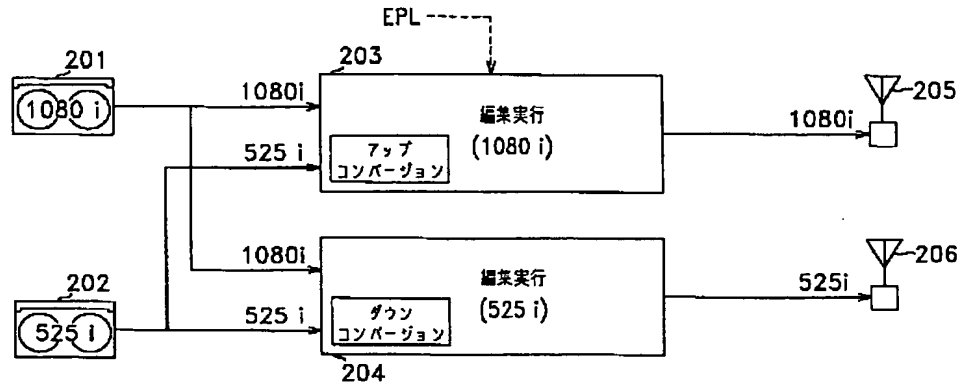


## 190 サイマルキャスト放送システム

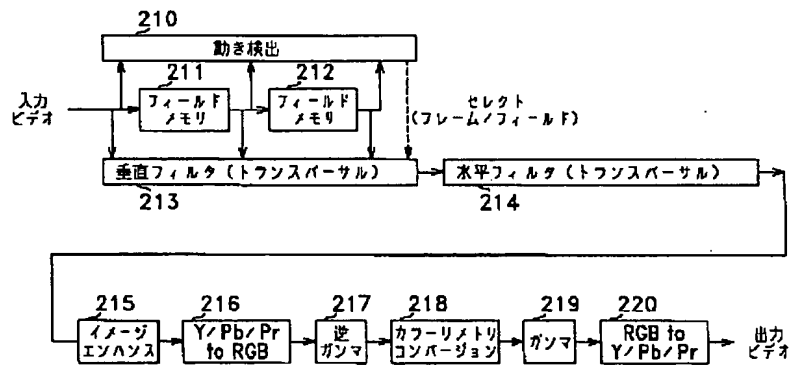


【図20】

200 サイマルキャスト放送システム



【図21】



フロントページの続き

(72)発明者 真貝 光俊  
東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内

(72)発明者 向 正孝  
東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内  
Fターム(参考) 5C053 FA14 FA21 GB21 JA16 JA21  
LA01 LA14